

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Michael MALCOLM

Serial No. N/Y/A (con. of serial no.  
09/127,249)

Filed: Herewith (parent filed July 31, 1998)

For: Multiple Cache Communication

Art Unit: Not Assigned

Examiner: Not Assigned

PRELIMINARY AMENDMENT

Dear Sir:

This is a preliminary amendment for a continuation application (filed herewith) under 37 C.F.R. § 1.53(b) of application serial no. 09/127,249 filed July 31, 1998 (now allowed).

Prior to examination, please amend the above-identified application as follows:

IN THE SPECIFICATION

Applicant respectfully requests that the specification be amended as indicated below. Marked-up pages showing these changes are also included.

- Please insert the following paragraph at page 1, line 9:

This application is a continuation of application serial no. 09/127,249, filed July 31, 1998 (now allowed).

- Please replace the paragraph beginning on page 7, line 18 with the following replacement paragraph:

As used herein, the terms “client” and “server” refer to relationships between the client or server and the cache, not necessarily to particular physical devices.

- Please replace the paragraph beginning on page 8, line 1 with the following replacement paragraph:

As used herein, the term “client device” includes any device taking on the role of a client in a client-server environment. There is no particular requirement that the client devices 120 must be individual devices; they can each be a single device, a set of cooperating devices, a portion of a device, or some combination thereof.

- Please replace the paragraph beginning on page 9, line 1 with the following replacement paragraph:

The server device 130 includes memory or storage 132 for recording one or more web objects 133. The web objects 133 can include any type of data suitable for transmitting to the client device 120, such as the following:

- Please replace the paragraph beginning on page 10, line 3 with the following replacement paragraph:

The cache system 110 includes a set of caches 111. The set of caches 111 comprises a variety of caches, preferably including root caches, leaf caches, intermediate caches and individual caches. Each cache 111 is designated a “leaf cache” if it is coupled to one or more client communication paths 121, and is designated a “root cache” if it is coupled to one or more server communication paths 131. The cache system 110 includes an inter-cache communication path 112 for communication between and among caches 111.

- Please replace the paragraph beginning on page 11, line 9 with the following replacement paragraph:

Each cache 111 includes a processor, program and data memory, and memory or storage 114 for recording one or more web objects 133. Each cache 111 retains the web objects 133 for repeated serving to client devices 120 in response to web requests.

- Please replace the paragraph beginning on page 12, line 6 with the following replacement paragraph:

If a client device 120 requests a cacheable web object 133, the leaf cache 111 might already have the requested web object 133 in its memory or storage 114. If so, the leaf cache 111 serves the requested web object 133 to the client device 120 without having to request the web object 133 from the root cache 111 or from the server device 130. If the leaf cache 111 does not already have the requested web object 133, the leaf cache 111 requests it from the root cache 111.

- Please replace the paragraph beginning on page 12, line 13 with the following replacement paragraph:

The root cache 111 performs a similar caching function, returning the requested cacheable web object 133 directly to the leaf cache 111 if it is already present in its own memory or storage 114, without having to request that web object 133 from the server device 130. If the root cache 111 does not already have the requested web object 133 in its memory or storage 114, the root cache 111 requests it from the server device 130.

- Please replace the paragraph beginning on page 12, line 20 with the following replacement paragraph:

If the leaf cache 111 and the root cache 111 do not already have a copy of the web object 133 in their respective memory or storage 114, the root cache 111 requests the web object 133 from the server device 120. Similarly, if the web object 133 is considered not cacheable, the root cache 111 requests the web object 133 from the server device 120 whether or not it has already that web object 133 in their respective memory or storage 114. The server device 130 receives the request and returns the requested web object 133 to the root cache 111.

- Please replace the paragraph beginning on page 13, line 8, with the following replacement paragraph:

The root cache 111 receives the requested web object 133 from the server device 130, records it in its memory or storage 114, and determines an object signature 134 for the web object 133. In a preferred embodiment, the root cache 111 computes the object signature 134 itself. In alternative embodiments, the server device 130 may compute and record the object signature 134 and transmit it to the root cache 111 with the web object 133.

- Please replace the paragraph beginning on page 13, line 19, with the following replacement paragraph:

There is no particular need for any device to be able to recover the web object 133 a priori from the object signature 134. It is sufficient that the root cache 111 or the leaf cache 111 can determine, in response to the object signature 134, if the web object 133 is present in its memory or storage 114, and if so, which web object 133 corresponds to that object signature 134.

- Please replace the paragraph beginning on page 14, line 4 with the following replacement paragraph:

If the web object 133 is cacheable but was requested from the server device 130, the request from the server device 130 was due to a cache miss. However, it can still occur that the leaf cache 111 (or some intermediate cache 111) already has the web objects 133 in its memory or storage 114, such as recorded in association with a different URL (uniform resource locator) or other identifier. In a preferred embodiment, each cache 111 records web objects 133 in association with the URL used to request those web objects 133.

- Please replace the paragraph beginning on page 14, line 12 with the following replacement paragraph:

For a first example, multiple server devices 130 can record mirror copies of identical web objects 133. For a second example, non-identical web

objects 133 can include identical embedded web objects 133 (such as common graphics, animation, or program fragments).

- Please replace the paragraph beginning on page 14 line 17 with the following replacement paragraph:

If the web object 133 is considered non-cacheable, it was requested from the server device 130 because non-cacheable web objects 133 are not meant to be served from the cache 111. However, it can still occur that the leaf cache 111 (or some intermediate cache 111) already has the web objects 133 in its memory or storage 114, because the non-cacheable web object 133 had been requested earlier.

- Please replace the paragraph beginning on page 15, line 7 with the following replacement paragraph:

The root cache 111 transmits the object signature 134 to the leaf cache 111. The leaf cache 111 determines, in response to the object signature 134, whether it already has the associated web object 133 in its memory or storage 114 and if so, which one is the associated web object 133. If so, the leaf cache 111 serves the associated web object 133 to the client device 120 from its memory or storage 114 without the root cache 111 having to actually transmit the entire web

object 133. If not, the root cache 111 transmits the actual web object 133 to the leaf cache 111, which can then serve it to the client device 120.

- Please replace the paragraph beginning on page 15, line 16 with the following replacement paragraph:

In a preferred embodiment, the root cache 111 includes a bitmap 115 in its memory or storage 114 for each non-cacheable web object 133, including one bit 116 for each leaf cache 111. Each bit 116 of the bitmap 115 indicates whether its associated leaf cache 111 has a copy of the web object 133.

- Please replace the paragraph beginning on page 15, line 21 and continuing on to page 16, line 6 with the following replacement paragraph:

The root cache 111 directly transmits the actual web object 133 to the leaf cache 111 if the associated bit 116 of the bitmap 115 indicates that the leaf does not have the web object 133. If the bit 116 indicates that the leaf cache 111 does have the web object 133, the root cache 111 attempts to transmit only the object signature 134. However, even if the bit 116 indicates that the leaf cache 111 does have the web object 133, it may occur that the leaf cache 111, being a cache, has discarded the web object 133 in the interim. In this case, the leaf cache 111 so indicates and re-requests the web object 133 from the root cache 111.



- Please replace the paragraph beginning on page 17, line 4 with the following replacement paragraph:

Those skilled in the art will recognize, after perusal of this application, that transmission of the object signature 134 in place of the actual web object 133 is a form of substantial compression. This form of compression is unreliable, in the computer science sense that the receiver is not guaranteed to be able to recover the web object 133 from its object signature 134. In fact, using this form of compression the leaf cache 111 can only do so if the web object 133 is already recorded in its memory or storage 114.

- Please replace the paragraph beginning on page 22, line 4 with the following replacement paragraph:

At a step 213, the leaf cache 111 determines if the web object 133 is present in its memory or storage 114. In a preferred embodiment, the leaf cache 111 makes this determination in response to the URL for the selected web object 133 included in the request from the client device 120. If the web object 133 is present, the method 200 proceeds with the next step. If the web object 133 is not present, the method 200 proceeds with the flow point 220.

- Please replace the paragraph beginning on page 22, line 14 with the following replacement paragraph:

At a flow point 220, the leaf cache 111 is unable to serve the web object 133 from its memory or storage 114, either because there has been a leaf cache miss or because the web object 133 is non-cacheable.

- Please replace the paragraph beginning on page 23, line 4 with the following replacement paragraph:

At a step 223, similar to the step 213, the root cache 111 determines if the web object 133 is present in its memory or storage 114. In a preferred embodiment, the root cache 111 makes this determination in response to the URL for the selected web object 133 included in the request from the client device 120. If the web object 133 is present, the method 200 proceeds with the next step. If the web object 133 is not present, the method 200 proceeds with the flow point 230.

- Please replace the paragraph beginning on page 23, line 14 with the following replacement paragraph:

At a flow point 230, the root cache 111 is unable to transmit the web object 133 from its memory or storage 114, either because there has been a root cache miss or because the web object 133 is non-cacheable.

- Please replace the paragraph beginning on page 24, line 7 with the following replacement paragraph:

At a step 234, the root cache 111 determines if the web object 133 is present in its memory or storage 114. In a preferred embodiment, the root cache 111 makes this determination in response to the object signature 134. If the web object 133 is present, the method 200 proceeds with the next step. If the web object 133 is not present, the method 200 proceeds with the flow point 240.

- Please replace the paragraph beginning on page 25, line 1 with the following replacement paragraph:

At a step 237, the leaf cache 111 determines if the web object 133 is present in its memory or storage 114, in response to the object signature 134. If the web object 133 is not present, the method 200 proceeds with the next step. If the web object 133 is present, the method 200 proceeds with the flow point 240.